Attention-Based Bidirectional Long Short-Term Memory Networks for Relation Classification

Peng Zhou, Wei Shi, Jun Tian, Zhenyu Qi[*], Bingchen Li, Hongwei Hao, Bo Xu
Institute of Automation, Chinese Academy of Sciences
{zhoupeng2013, shiwei2013, tianjun2013, zhenyu.qi,
libingchen2013, hongwei.hao, xubo}@ia.ac.cn

Abstract

Relation classification is an important 语义的 semantic processing task in the field of natural language processing (NLP). State-of-the-art systems still rely on 词汇的 lexical resources such as WordNet or NLP systems like 依赖性解析器 dependency parser and 命名实体识别器 named entity recognizers (NER) to get high-level features. Another challenge is that important information can appear at any position in the sentence. To tackle these problems, we propose Attention-Based Bidirectional Long Short-Term Memory Networks(Att-BLSTM) to capture the most important semantic information in a sentence. The experimental results on the SemEval-2010 relation classification task show that our method outperforms most of the existing methods, with only word vectors.

1 Introduction

关系分类 Relation classification is the task of finding 语义的 semantic relations between pairs of 名词性词 nominals, which is useful for many NLP applications, such as information extraction (Wu and Weld, 2010), question answering (Yao and Van Durme, 2014). For instance, the following sentence contains an example of the 实体目标的关系 Entity-Destination relation between the nominals **Flowers** and **chapel**. 小教堂

$\langle e_1 \rangle$ **Flowers** $\langle /e_1 \rangle$ are carried into the $\langle e_2 \rangle$ **chapel** $\langle /e_2 \rangle$.

$\langle e_1 \rangle$, $\langle /e_1 \rangle$, $\langle e_2 \rangle$, $\langle /e_2 \rangle$ are four position indicators which specify the starting and ending of the nominals (Hendrickx et al., 2009).

Traditional relation classification methods that 采用从 employ handcrafted features from lexical resources, are usually based on pattern matching, and have achieved high performance (Bunescu and Mooney, 2005; Mintz et al., 2009; Rink and Harabagiu, 2010). One downside of these methods is that many traditional NLP systems are utilized to extract high-level features, such as part of speech tags, shortest dependency path and named entities, which consequently results in the increase of computational cost and additional propagation errors. Another downside is that designing features manually is time-consuming, and performing poor on generalization due to the low coverage of different training datasets.

之前模型的缺点

Recently, deep learning methods provide an effective way of reducing the number of handcrafted features (Socher et al., 2012; Zeng et al., 2014). However, these approaches still use lexical resources such as WordNet (Miller, 1995) or NLP systems like dependency parsers and NER to get high-level features.

This paper proposes a novel neural network Att-BLSTM for relation classification. Our model utilizes neural attention mechanism with Bidirectional Long Short-Term Memory Networks(BLSTM) to capture the most important semantic information in a sentence. This model doesn't utilize any features derived from lexical resources or NLP systems.

The contribution of this paper is using BLSTM with attention mechanism, which can automatically focus on the words that have 决定性的 decisive effect on classification, to capture the most important semantic information in a sentence, without using extra knowledge and NLP systems. We conduct experiments on the SemEval-2010 Task 8 dataset, and achieve an $F1$-score of 84.0%, higher than most of the existing methods in the literature.

The remainder of the paper is structured as follows. In Section 2, we review related work about relation classification. Section 3 presents our Att-BLSTM model in detail. In Section 4, we describe details about the setup of experimental evaluation

[*] Correspondence author: zhenyu.qi@ia.ac.cn

Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, pages 207–212,
Berlin, Germany, August 7-12, 2016. ©2016 Association for Computational Linguistics

and the experimental results. Finally, we have our conclusion in Section 5.

2 Related Work

Over the years, various methods have been proposed for relation classification. Most of them are based on pattern matching and apply extra NLP systems to derive lexical features. One related work is proposed by Rink and Harabagiu (2010), which utilizes many features derived from external corpora for a Support Vector Machine(SVM) classifier.

Recently, deep neural networks can learn underlying features automatically and have been used in the literature. Most representative progress was made by Zeng et al. (2014), who utilized convolutional neural networks(CNN) for relation classification. While CNN is not suitable for learning long-distance semantic information, so our approach builds on Recurrent Neural Network(RNN) (Mikolov et al., 2010).

One related work was proposed by Zhang and Wang (2015), which employed bidirectional RNN to learn patterns of relations from raw text data. Although bidirectional RNN has access to both past and future context information, the range of context is limited due to the vanishing gradient problem. To overcome this problem, Long short-Term memory(LSTM) units are introduced by Hochreiter and Schmidhuber (1997).

Another related work is SDP-LSTM model proposed by Yan et al. (2015). This model leverages the shortest dependency path(SDP) between two nominals, then it picks up 多样化的 heterogeneous information along the SDP with LSTM units. While our method regards the raw text as a sequence.

Finally, our work is related to BLSTM model proposed by Zhang et al. (2015). This model utilizing NLP tools and lexical resources to get word, position, POS, NER, dependency parse and hypernym features, together with LSTM units, achieved a comparable result to the state-of-the-art. However, comparing to the complicated features that employed by Zhang et al. (2015), our method regards the four position indicators $\langle e1\rangle, \langle/e1\rangle, \langle e2\rangle, \langle/e2\rangle$ as single words, and transforms all words to word vectors, forming a simple but competing model.

3 Model

In this section we propose Att-BLSTM model in detail. As shown in Figure 1, the model proposed in this paper contains five components:

(1) Input layer: input sentence to this model;

(2) Embedding layer: map each word into a low dimension vector;

(3) LSTM layer: utilize BLSTM to get high level features from step (2);

(4) Attention layer: produce a weight vector, and merge word-level features from each time step into a sentence-level feature vector, by multiplying the weight vector;

(5) Output layer: the sentence-level feature vector is finally used for relation classification.

These components will be presented in detail in this section.

3.1 Word Embeddings

Given a sentence consisting of T words $S = \{x_1, x_2, \ldots, x_T\}$, every word x_i is converted into a real-valued vector e_i. For each word in S, we first look up the embedding matrix $W^{wrd} \in \mathbb{R}^{d^w\,|V|}$, where V is a fixed-sized vocabulary, and d^w is the size of word embedding. The matrix W^{wrd} is a parameter to be learned, and d^w is a hyper-parameter to be chosen by user. We transform a word x_i into its word embedding e_i by using the matrix-vector product:

$$e_i = W^{wrd} v^i \tag{1}$$

where v^i is a vector of size $|V|$ which has value 1 at index e_i and 0 in all other positions. Then the sentence is feed into the next layer as a real-valued vectors $emb_s = \{e_1, e_2, \ldots, e_T\}$.

3.2 Bidirectional Network

LSTM units are firstly proposed by Hochreiter and Schmidhuber (1997) to overcome gradient vanishing problem. The main idea is to introduce an adaptive gating mechanism, which decides the degree to which LSTM units keep the previous state and memorize the extracted features of the current data input. Then lots of LSTM variants have been proposed. We adopt a variant introduced by Graves et al. (2013), which adds weighted peephole connections from the *Constant Error Carousel* (CEC) to the gates of the same memory block. By directly employing the current cell state to generate the gate degrees, the 门镜 peephole connections allow all gates to 进行视察 *inspect* into the cell (i.e.

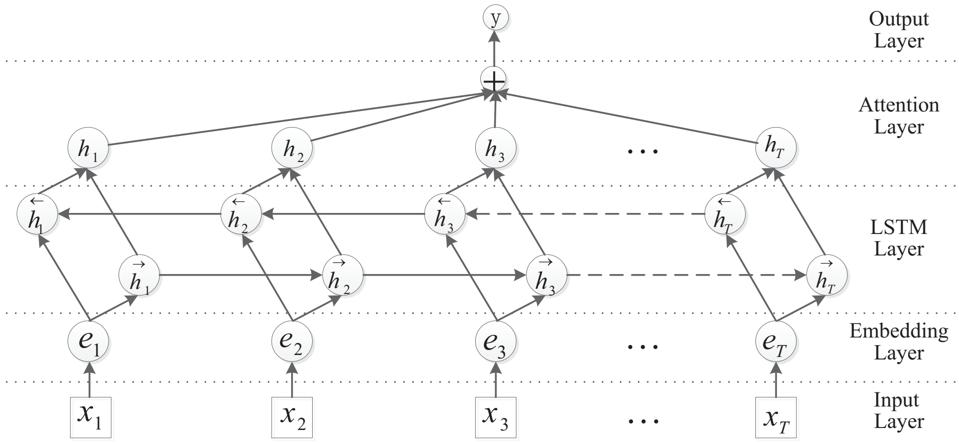

Figure 1: Bidirectional LSTM model with Attention

the current cell state) even when the output gate is closed (Graves, 2013).

LSTM 的 结 构

Typically, four components composite the LSTM-based recurrent neural networks: one input gate (2) i_t with corresponding weight matrix $W_{xi}, W_{hi}, W_{ci}, b_i$; one (3) forget gate f_t with corresponding weight matrix $W_{xf}, W_{hf}, W_{cf}, b_f$; one (6) output gate o_t with corresponding weight matrix $W_{xo}, W_{ho}, W_{co}, b_o$, all of those gates are set to generate some degrees, using current input x_i, the state h_{i-1} that previous step generated , and current state of this cell c_{i-1} (peephole), for the decisions whether to take the inputs, forget the memory stored before, and output the state generated later. Just as these following equations demonstrate:

$$i_t = \sigma(W_{xi}x_t + W_{hi}h_{t-1} + W_{ci}c_{t-1} + b_i) \quad (2)$$

$$f_t = \sigma(W_{xf}x_t + W_{hf}h_{t-1} + W_{cf}c_{t-1} + b_f) \quad (3)$$

$$g_t = \tanh(W_{xc}x_t + W_{hc}h_{t-1} + W_{cc}c_{t-1} + b_c) \quad (4)$$

$$c_t = i_t g_t + f_t c_{t-1} \quad (5)$$

$$o_t = \sigma(W_{xo}x_t + W_{ho}h_{t-1} + W_{co}c_t + b_o) \quad (6)$$

$$h_t = o_t \tanh(c_t) \quad (7)$$

Hence, current cell state c_t will be generated by calculating the weighted sum using both previous cell state and current information generated by the cell (Graves, 2013).

For many sequence modelling tasks, it is beneficial to have access to future as well as past context. However, standard LSTM networks process sequences in temporal order, they ignore future context. Bidirectional LSTM networks extend the unidirectional LSTM networks by introducing a second layer, where the hidden to hidden connections flow in opposite temporal order. The model is therefore able to exploit information both from the past and the future.

In this paper, we use BLSTM. As also shown in Figure 1, the network contains two sub-networks for the left and right sequence context, which are forward and backward pass respectively. The output of the i^{th} word is shown in the following equation:

$$h_i = [\overrightarrow{h_i} \oplus \overleftarrow{h_i}] \quad (8)$$

元素方式的加和

Here, we use element-wise sum to combine the forward and backward pass outputs.

3.3 Attention

Attentive neural networks have recently demonstrated success in a wide range of tasks ranging from question answering, machine translations, speech recognition, to image captioning (Hermann et al., 2015; Bahdanau et al., 2014; Chorowski et al., 2015; Xu et al., 2015). In this section, we propose the attention mechanism for relation classification tasks. Let H be a matrix consisting of output vectors$[h_1, h_2, \ldots, h_T]$ that the LSTM layer produced, where T is the sentence length. The representation r of the sentence is formed by a weighted sum of these output vectors:

$$M = \tanh(H) \quad (9)$$

$$\alpha = softmax(w^T M) \quad (10)$$

$$r = H\alpha^T \quad (11)$$

Model	Feature Set	F1
SVM (Rink and Harabagiu, 2010)	POS, prefixes, morphological, WordNet, dependency parse, Levin classed, ProBank, FramNet, NomLex-Plus, Google n-gram, paraphrases, TextRunner	82.2
CNN (Zeng et al., 2014)	WV (Turian et al., 2010) (dim=50) + PF + WordNet	69.7 82.7
RNN (Zhang and Wang, 2015)	WV (Turian et al., 2010) (dim=50) + PI WV (Mikolov et al., 2013) (dim=300) + PI	80.0 82.5
SDP-LSTM (Yan et al., 2015)	WV (pretrained by word2vec) (dim=200), syntactic parse + POS + WordNet + grammar relation embeddings	82.4 83.7
BLSTM (Zhang et al., 2015)	WV (Pennington et al., 2014) (dim=100) + PF + POS + NER + WNSYN + DEP	82.7 84.3
BLSTM Att-BLSTM BLSTM Att-BLSTM	WV (Turian et al., 2010) (dim=50) + PI WV (Turian et al., 2010) (dim=50) + PI WV (Pennington et al., 2014) (dim=100) + PI WV (Pennington et al., 2014) (dim=100) + PI	80.7 82.5 82.7 84.0

Table 1: Comparison with previous results. WV, PF, PI stand for word vectors, position features and position indicators respectively.

where $H \in \mathbb{R}^{d^w \times T}$, d^w is the dimension of the word vectors, w is a trained parameter vector and w^{T} is a transpose. The dimension of w, α, r is d^w, T, d^w separately.

We obtain the final sentence-pair representation used for classification from:

$M = \tanh(H)$

$\alpha = softmax(w^T M)$

$r = H\alpha^T$

$$h^* = \tanh(r) \tag{12}$$

3.4 Classifying

In this setting, we use a softmax classifier to predict label $\hat{y}$ from a discrete set of classes Y for a sentence S. The classifier takes the hidden state h^* as input:

$$\hat{p}\left(y|S\right) = softmax\left(W^{(S)}h^* + b^{(S)}\right) \tag{13}$$

$$\hat{y} = \arg\max_{y} \hat{p}\left(y|S\right) \tag{14}$$

The cost function is the negative log-likelihood of the true class labels $\hat{y}$:

$$J\left(\theta\right) = -\frac{1}{m}\sum_{i=1}^{m} t_i \log(y_i) + \lambda\|\theta\|_F^2 \tag{15}$$

where $\boldsymbol{t} \in \Re^m$ is the one-hot represented ground truth and $\boldsymbol{y} \in \Re^m$ is the estimated probability for each class by softmax (m is the number of target classes), and λ is an L2 regularization hyperparameter. In this paper, we combine dropout with L2 regularization to alleviate overfitting.

3.5 Regularization

Dropout, proposed by (Hinton et al., 2012), prevents co-adaptation of hidden units by randomly omitting feature detectors from the network during forward propagation. We employ dropout on the embedding layer, LSTM layer and the penultimate layer.

We additionally constrain $L2$-norms of the weight vectors by rescaling w to have $\|w\| = s$, whenever $\|w\| > s$ after a gradient descent step, as shown in equation 15. Training details are further introduced in Section 4.1.

4 Experiments

4.1 Dataset and Experimental Setup

Experiments are conducted on SemEval-2010 Task 8 dataset (Hendrickx et al., 2009). This dataset contains 9 relationships (with two directions) and an undirected Other class. There are 10,717 annotated examples, including 8,000 sentences for training, and 2,717 for testing. We adopt the official evaluation metric to evaluate our systems, which is based on macro-averaged F1-score for the nine actual relations (excluding the Other relation) and takes the directionality into consideration.

In order to compare with the work by Zhang and Wang (2015), we use the same word vectors proposed by Turian et al. (2010) (50-dimensional) to initialize the embedding layer. Additionally, to compare with the work by Zhang et al. (2015), we also use the 100-dimensional word vectors pre-trained by Pennington et al. (2014).

Since there is no official development dataset, we randomly select 800 sentence for validation. The hyper-parameters for our model were tuned on the development set for each task. Our model was trained using AdaDelta (Zeiler, 2012) with a learning rate of 1.0 and a minibatch size 10. The model parameters were regularized with a per-minibatch L2 regularization strength of 10^{-5}. We evaluate the effect of dropout embedding layer, dropout LSTM layer and dropout the penultimate layer, the model has a better performance, when the dropout rate is set as 0.3, 0.3, 0.5 respectively. Other parameters in our model are initialized randomly.

4.2 Experimental Results

Table 1 compares our Att-BLSTM with other state-of-the-art methods of relation classification.

SVM: This is the top performed system in SemEval-2010. Rink and Harabagiu (2010) leveraged a variety of handcrafted features, and use SVM as the classifier. They achieved an F_1-score of 82.2%.

CNN: Zeng et al. (2014) treated a sentences as a sequential data and exploited the convolutional neural network to learn sentence-level features; they also used a special position vector to represent each word. Then the sentence-level and lexical features were concatenated into a single vector and fed into a softmax classifier for prediction. This model achieves an F_1-score of 82.7%.

RNN: Zhang and Wang (2015) employed bidirectional RNN networks with two different dimension word vectors for relation classification. They achieved an F_1-score of 82.8% using 300-dimensional word vectors pre-trained by Mikolov et al. (2013), and an F_1-score of 80.0% using 50-dimensional word vectors pre-trained by Turian et al. (2010). Our model with the same 50-dimensional word vectors achieves an F_1-score of 82.5%, about 2.5 percent more than theirs.

SDP-LSTM: Yan et al. (2015) utilized four different channels to pick up heterogeneous along the SDP, and they achieved an F_1-score of 83.7%. Comparing with their model, our model regarding the raw text as a sequence is simpler.

BLSTM: Zhang et al. (2015) employed many features derived from NLP tools and lexical resources with bidirectional LSTM networks to learn the sentence level features, and they achieved state-of-the-art performance on the SemEval-2010 Task 8 dataset. Our model with the same word vectors achieves a very similar result (84.0%), and our model is more simple.

Our proposed Att-BLSTM model yields an F_1-score of 84.0%. It outperforms most of the existing competing approaches, without using lexical resources such as WordNet or NLP systems like dependency parser and NER to get high-level features.

5 Conclusion

In this paper, we propose a novel neural network model, named Att-BLSTM, for relation classification. This model does not rely on NLP tools or lexical resources to get, it uses raw text with position indicators as input. The effectiveness of Att-BLSTM is demonstrated by evaluating the model on SemEval-2010 relation classification task.

Acknowledgments

This research was supported by the National High Technology Research and Development Program of China (No.2015AA015402). We thank the anonymous reviewers for their insightful comments.

References

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*.

Razvan C Bunescu and Raymond J Mooney. 2005. A shortest path dependency kernel for relation extraction. In *Proceedings of the conference on human language technology and empirical methods in natural language processing*, pages 724–731. Association for Computational Linguistics.

Jan K Chorowski, Dzmitry Bahdanau, Dmitriy Serdyuk, Kyunghyun Cho, and Yoshua Bengio. 2015. Attention-based models for speech recognition. In *Advances in Neural Information Processing Systems*, pages 577–585.

Alan Graves, Abdel-rahman Mohamed, and Geoffrey Hinton. 2013. Speech recognition with deep recurrent neural networks. In *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*, pages 6645–6649. IEEE.

Alex Graves. 2013. Generating sequences with recurrent neural networks. *arXiv preprint arXiv:1308.0850*.

Iris Hendrickx, Su Nam Kim, Zornitsa Kozareva, Preslav Nakov, Diarmuid Ó Séaghdha, Sebastian Padó, Marco Pennacchiotti, Lorenza Romano, and Stan Szpakowicz. 2009. Semeval-2010 task 8: Multi-way classification of semantic relations between pairs of nominals. In *Proceedings of the Workshop on Semantic Evaluations: Recent Achievements and Future Directions*, pages 94–99. Association for Computational Linguistics.

Karl Moritz Hermann, Tomas Kocisky, Edward Grefenstette, Lasse Espeholt, Will Kay, Mustafa Suleyman, and Phil Blunsom. 2015. Teaching machines to read and comprehend. In *Advances in Neural Information Processing Systems*, pages 1684–1692.

Geoffrey E Hinton, Nitish Srivastava, Alex Krizhevsky, Ilya Sutskever, and Ruslan R Salakhutdinov. 2012. Improving neural networks by preventing co-adaptation of feature detectors. *arXiv preprint arXiv:1207.0580*.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation*, 9(8):1735–1780.

Tomas Mikolov, Martin Karafiát, Lukas Burget, Jan Cernockỳ, and Sanjeev Khudanpur. 2010. Recurrent neural network based language model. In *INTERSPEECH 2010, 11th Annual Conference of the International Speech Communication Association, Makuhari, Chiba, Japan, September 26-30, 2010*, pages 1045–1048.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119.

George A Miller. 1995. Wordnet: a lexical database for english. *Communications of the ACM*, 38(11):39–41.

Mike Mintz, Steven Bills, Rion Snow, and Dan Jurafsky. 2009. Distant supervision for relation extraction without labeled data. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP: Volume 2-Volume 2*, pages 1003–1011. Association for Computational Linguistics.

Jeffrey Pennington, Richard Socher, and Christopher D Manning. 2014. Glove: Global vectors for word representation. In *EMNLP*, volume 14, pages 1532–1543.

Bryan Rink and Sanda Harabagiu. 2010. Utd: Classifying semantic relations by combining lexical and semantic resources. In *Proceedings of the 5th International Workshop on Semantic Evaluation*, pages 256–259. Association for Computational Linguistics.

Richard Socher, Brody Huval, Christopher D Manning, and Andrew Y Ng. 2012. Semantic compositionality through recursive matrix-vector spaces. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 1201–1211. Association for Computational Linguistics.

Joseph Turian, Lev Ratinov, and Yoshua Bengio. 2010. Word representations: a simple and general method for semi-supervised learning. In *Proceedings of the 48th annual meeting of the association for computational linguistics*, pages 384–394. Association for Computational Linguistics.

Fei Wu and Daniel S Weld. 2010. Open information extraction using wikipedia. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 118–127. Association for Computational Linguistics.

Kelvin Xu, Jimmy Ba, Ryan Kiros, Aaron Courville, Ruslan Salakhutdinov, Richard Zemel, and Yoshua Bengio. 2015. Show, attend and tell: Neural image caption generation with visual attention. *arXiv preprint arXiv:1502.03044*.

Xu Yan, Lili Mou, Ge Li, Yunchuan Chen, Hao Peng, and Zhi Jin. 2015. Classifying relations via long short term memory networks along shortest dependency path. *arXiv preprint arXiv:1508.03720*.

Xuchen Yao and Benjamin Van Durme. 2014. Information extraction over structured data: Question answering with freebase. In *ACL (1)*, pages 956–966. Citeseer.

Matthew D Zeiler. 2012. Adadelta: An adaptive learning rate method. *arXiv preprint arXiv:1212.5701*.

Daojian Zeng, Kang Liu, Siwei Lai, Guangyou Zhou, and Jun Zhao. 2014. Relation classification via convolutional deep neural network. In *Proceedings of COLING*, pages 2335–2344.

Dongxu Zhang and Dong Wang. 2015. Relation classification via recurrent neural network. *arXiv preprint arXiv:1508.01006*.

Shu Zhang, Dequan Zheng, Xinchen Hu, and Ming Yang. 2015. Bidirectional long short-term memory networks for relation classification.